Logical Data Interface

The IBM 3830-2 is a programmable storage control unit. It maintains tables that reflect the contents and utilization of direct access storage device (DASD) volumes attached to it. It also controls the movement of data from an attached cartridge library to the DASD volumes. The 3830-2 is thus an instance of "outboard intelligence"; i.e., a programmable hardware component that is exterior to the CPU. Described below is a proposal for the use of such a hardware component (hereinafter called an "Outboard Data Manager - ODM") to provide a new, higher level I/O interface for system 370 CPUs.

The interface between the CPU and the ODM is a logical interface. The interface is logical in that the information passed by the CPU does not specify the physical location of the data being requested. The physical location of the data is the responsibility of, and is known only by, the ODM. The variables used by the CPU to identify the requested data are a Data Set Identifier and a Record Identifier.

The Data Set Identifier (DSID) is a unique six-byte binary number generated by the ODM when a data set is created. The operating system within the CPU retains this value and passes it to the ODM with each request involving the data set. The generation of the DSID is performed by the ODM rather than the CPU, in order to permit the ODM to be attached to multiple independent CPUs. The centralization of the generation function in the ODM guarantees uniqueness of DSIDs across multiple CPUs.

The Record Identifier (RID) is a six-byte binary number. This value is actually the logical name of a physical block within a data set. The generation of this value is under the control of the creator of the data set within the operating system.

The DSID and the RID are, therefore, the basic components of the ODM interface. The operating system within the CPU uses these values to communicate with the ODM via the following commands:

1) ASSIGN DSID
   This command is issued by the operating system whenever it is about to create a new data set. The ODM maintains a six-byte DSID counter. Upon receiving this command, the ODM returns the current value of the counter to the requestor and increments the counter value by one.
2) SET DSID (-RID)
   This command is used to establish a context for a subsequent (through CCW command chaining) ATTRIBUTE, RETRIEVE, WRITE, READ, or DELETE-RECORD command. The SET command identifies the DSID to be associated with a ATTRIBUTE or RETRIEVE command. It identifies the DSID and RID of the record to be read or written by a subsequent READ or WRITE command.
3) ATTRIBUTE.
   The ATTRIBUTE command is used by the operating system to pass information about a data set (data set attributes) to the ODM. The command is issued after a DSID for the data set has been

assigned by the ODM (after an ASSIGN DSID has been done). The data set with which the command is associated is identified by the DSID value specified in the preceding SET command. The attributes which may be associated with the data set are as follows:

a. Record Size - the size in bytes of the records within the data set.

b. Maximum RID - a binary number from 0 - 2/48/ - 1. This value determines the maximum value of an RID for this DSID.

c. Frequency of Access - a binary number from 0 - 255. The value indicates the rate at which the operating system expects to access the data set. A value of 255 indicates very frequent access; a value of 0 indicates an extremely low access rate.

Record size is chosen as a data set attribute to enforce fixed length records. In a relocate system it is assumed that the record length would be equal to the page size.

The frequency of access value serves two purposes:

(a) If the storage controlled by the ODM is of several different types, with correspondingly different access rates, this value may assist the ODM in determining the type of storage to use for this DSID. The more frequent the access, the higher the access rate of the storage should be.

(b) If the ODM controls more than one physical device, this value will assist it in balancing the load across the devices.

An ATTRIBUTE command that defines the size of the records in the data set must be issued prior to writing the first record in the data set. Once defined, this value may not be altered. The "frequency of access" and "maximum RID" may be altered at any time during the life of the data set.

4) WRITE

This command is used to transfer a data record from main storage to the storage controlled by the ODM. The data set of which the record is a part and the "name" of the record are identified, by the DSID and RID operands of a preceding SET command.

The WRITE command is used to write new records (new RID values) into the data set and to update existing records (previously written RID values). The specification of a new RID value causes the ODM to establish an association between the data record and the RID value.

5) READ.

This command is used to transfer a data record from the storage controlled by the ODM to main storage. The data set which

contains the record and the name of the record are identified, by the DSID and RID values specified in the preceding SET command.

6) RETRIEVE.
This command is used to cause the ODM to return information about a data set. The data set about which information is returned is identified by the DSID value specified in the preceding SET command. The following information is returned by
the ODM:
a. High RID - The current largest RID value associated with a record in the data set.
b. Maximum RID - The largest RID that is valid for this DSID.
c. Record size - The size of the records within the data set.
d. Frequency of Access Value - The current "frequency of access value associated with the data set".

7) DELETE-RECORD
This command causes the ODM to free the storage occupied by the record(s) associated with the specified RID value(s). The data set in which the records exist are identified by the preceding SET command. Subsequent READ commands for records that
have been deleted cause the ODM to return a "no record found" indication.

8) DELETE-DATASET
This command causes the ODM to delete a data set. Upon receipt of the command, the ODM frees all storage occupied by records of the data set and destroys all traces of the DSID. Subsequent references to a destroyed DSID causes the ODM to return a "nonexistent DSID" indication.

The use of the Logical Data Interface would improve the performance of a relocate system. The major performance improvement is a result of a reduction in the amount of CPU overhead associated with I/O requests. The channel programs are simpler; hence their virtual to real translation requires many fewer instructions. In addition, the creation of the channel programs is a much simpler process, in that they are no longer dependent on either the physical location of the data or the physical characteristics of the storage media.

An additional performance improvement, again in terms of reduced CPU overhead, is realized by using this interface for paging. The logical data interface lends itself naturally to the paging algorithms of the VS systems. In a single virtual memory system, all pages would be contained within one DSID; an RID would exist for each virtual page. Calculation of the RID is performed by simply dividing the virtual address by the page size and ignoring the remainder. In a multiple virtual memory system each virtual memory would occupy a unique DSID. RIDs for pages in the "private area" are again calculated by dividing the virtual address by the page size and ignoring the remainder. The Logical Data Interface reduces CPU overhead, by eliminating the requirement to invoke another software

component to convert the DSID and RID to an actual physical location; the new interface permits the DSID and RID to be passed directly to the ODM.

The use of the Logical Data Interface results in device independent software. The operating system is totally independent of the storage controlled by the ODM and the physical characteristics of that storage. Thus, the physical characteristics of that storage (for example, the track length of a device) can be changed without impacting the operating system. This provides a major reduction in the development cost of new hardware products. The software cost typically required to support a new device would be eliminated; no software changes would be required. In addition, since software support would not be needed, shipment of a new device would be independent of any software release. Finally, software device independence guarantees that new products would be concurrently supported by the different operating systems, and all versions of them, when the product is shipped.

The Logical Data Interface provides more efficient-usage of direct-access storage. In a loosely coupled CPU environment, it would be possible for the multiple CPUs to use the ODM storage for paging and spooling. (The RESERVE/RELEASE overhead that would be required to achieve this with the current DASD interface is prohibitive.) In addition, since space allocation can be performed by the ODM as the records are written, a data set will occupy only as much space as it requires. The requirement to preallocate a larger amount of space normally imposed by operating systems, is eliminated.

The Logical Data Interface provides improved integrity. This is done in two ways. First, having restricted a user to a given data set (DSID), it is impossible for him to generate an RID value that results in the reading of a record from another data set. The Logical Data Interface also makes it impossible to read data that was in a previously deleted data set, because the same storage is being used for a new data set. The interface prevents this by requiring an RID to be written before it can be read.